Amendments to the Claims:

Please amend claims 1, 3, 5, 9 and 10, and please add new claims 13 and 14 as follows. This listing of claims replaces all prior versions, and listings, of claims in the application.

Listing of claims:

1. (currently amended) A branch predictor for a multi-processing computer, able to execute multiple processes, each process having a designated process reference, comprising:

a global history register for storing a branch history of previous sequential branch instructions for a plurality of the multiple processes;

a hash logic for creating an index from a combination of a process reference of a process corresponding to a current branch instruction, an address of the current branch instruction, and the branch history for the plurality of the multiple processes;

a branch prediction table for storing branch prediction reference data, and for outputting branch prediction reference data corresponding to the index created by the hash logic;

an address selection circuit for selecting one of a target address known from the current branch instruction and a next ~~instruction~~ address of the current branch instruction to generate a branch prediction address, in response to the branch prediction reference data output from the branch prediction table; and

a branch prediction result tester generating a control signal for updating the branch history stored in the global history register and the branch prediction reference data stored in the branch prediction table, in response to a comparison between a real branch address and the branch prediction address ~~according to an execution result of the current branch instruction.~~ wherein the address selection circuit generates the branch prediction address further in response to a state of the control signal generated by the branch prediction result tester.

2. (original) The branch predictor of claim 1, wherein the branch prediction table comprises a plurality of up/down saturating counters selected by the index created by the hash

logic.

3. (currently amended) The branch predictor of claim 1, wherein the global history register comprises a shift register.

4. (previously presented) The branch predictor of claim 1, wherein the process reference comprises a process ID corresponding to the current branch instruction, and wherein the hash logic creates the index by performing an exclusive-OR operation on the process ID corresponding to the current branch instruction, the address of the current branch instruction, and the branch history.

5. (currently amended) The branch predictor of claim 1, wherein the branch prediction result tester includes a comparator for determining whether [[a]]the real branch address ~~according to the execution result of the current branch instruction~~ matches with the branch prediction address, and creates [[a]]the control signal ~~corresponding to the result~~.

6. (original) The branch predictor of claim 5, wherein the comparator generates a control signal of logic "1" if the real branch address matches with the branch prediction address, and generates a control signal of logic "0" if the real branch address does not match.

7. (previously presented) The branch predictor of claim 6, wherein the address selection circuit changes and outputs the real branch address as the branch prediction address when the control signal is logic "0".

8. (previously presented) The branch predictor of claim 6, wherein the branch prediction table comprises an up/down counter, and wherein the up/down counter increments when the control signal is logic "1", and wherein the up/down counter decrements when the control signal is logic "0".

9. (currently amended) The branch predictor of claim 6, wherein the global history register comprises a shift register, and wherein the shift register shifts the branch prediction result in a first direction by inserting the control signal.

10. (currently amended) A method of predicting a branch address of a conditional branch instruction with reference to a branch prediction table for storing branch prediction reference data in a multi-processing computer able to execute multiple processes, each having a designated process ID, the method comprising the steps of:

creating an index to access the branch prediction table from a combination of a process ID of a process corresponding to the conditional branch instruction, an address of the conditional branch instruction, and a branch history comprising previous sequential branch instructions for a plurality of the multiple processes;

reading branch prediction reference data from the branch prediction table in response to the index;

selectively outputting one of a target address known from the conditional branch instruction and a next address of the conditional branch instruction to generate a branch prediction address, in response to the branch prediction reference data, and further in response to a state of a control signal; [[and]]

generating the control signal in response to a comparison between a real branch address and the branch prediction address; and

updating the branch history and the stored branch prediction reference data in the branch prediction table in response to the control signal ~~a real branch address according to an execution result of the conditional branch instruction~~.

11. (previously presented) The method of claim 10 further comprising the steps of:

determining whether the real branch address matches with the branch prediction address; and

changing and outputting a corrected branch address as the branch prediction address if the real branch address does not match therewith.

12. (previously presented) The method of claim 10, wherein creating the index comprises performing an exclusive-OR operation on the process ID of the process corresponding to the conditional branch instruction, the address of the conditional branch instruction, and the branch history comprising previous sequential branch instructions.

13. (new) The branch predictor of claim 1, wherein the address selection circuit generates the branch prediction address when the control signal is in a first state, and wherein the address selection circuit performs no operation when the control signal is in a second state.

14. (new) The method of claim 10, wherein the branch prediction address is generated when the control signal is in a first state, and wherein the branch prediction address is not generated when the control signal is in a second state.

Amendments to the Drawings:

The attached sheet of drawings includes changes to Fig. 2. This sheet, which includes Fig. 2, replaces the original sheet.

A marked-up version of the drawings, with revisions shown in red, is included with the amended drawings. Entry of the amended drawings is respectfully requested.

Approved BSO 8/11/04

Attachment: Replacement Sheet

Annotated Sheet Showing Changes